Monte-Carlo and Empirical Methods for Statistical Inference, FMS091/MA3111

L13 — Summary and comments

Inversion and transformation methods

Most standard computing languages have functions for generating uniform pseudo-random numbers. Assuming that we have access to \( U(0,1) \), we can generate random numbers with other distributions in different ways:

- **Inversion method**: To generate \( x \in \mathbb{R} \) with a continuous distribution \( F \), take \( x = F^{-1}(u) \), where \( u \in U(0,1) \).
- **Transformation theorem**: If \( X \) has a density \( f \), then the density of \( Z = g(X) \) is \( f(h(z))|J| \), where \( h = g^{-1} \) and \( J \) is the Jacobian determinant.
- **If are \( X \) and \( Y \) independent random variables with densities \( f_x \) and \( f_y \), then the density of \( Z = X + Y \) is \( f_z(z) = \int f_x(z-t)f_y(t) \, dt \).
Markov Chain Monte Carlo

- Basic idea: To sample from a density $f$ we construct a Markov chain with $f$ as its stationary distribution.
- If $f$ is complex and/or high dimensional this is often easier than transformation methods or rejection sampling.
- But, the samples will not be independent.
- Start with a Markov chain with transition kernel $q(x,y)$.
- Accept the jumps with some probability $\alpha(x,y)$ chosen such that the combined Markov chain has the correct stationary distribution.
- Taking
  \[ \alpha(x,y) = \min\left(1, \frac{f(y)q(y,x)}{f(x)q(x,y)}\right), \]
  ensures detailed balance.

Different proposals

There are a number of different ways of constructing the proposal kernel, $q(x,y)$,

- Independent proposals, $q(x,y) = q(y)$. Draw new points $y$ independent of the current state $x$.
- Symmetric proposals, $q(x,y) = q(y,x)$. Commonly this is $y = x + \varepsilon$ with $\varepsilon$ as
  - $\varepsilon \sim \mathcal{N}(0,\Sigma)$, random walk proposal.
  - $\varepsilon \sim \mathcal{U}(-R,R)$.
- Multiplicative proposals, $y = x\varepsilon$. Typically used for skewed distributions, such as variance parameters.
- and others.

The Metropolis Hastings algorithm

Algorithm:
Given a density $f(x)$ and a proposal kernel $q(x,y)$
Start the chain with some $x^{(0)}$, and loop over $t = 1, \ldots, T$.
1. Given $x^{(t)}$, draw a proposal $y$ from $q(x^{(t)},y)$.
2. Calculate the acceptance probability
   \[ \alpha(x^{(t)},y) = \min\left(1, \frac{f(y)q(y,x^{(t)})}{f(x^{(t)})q(x^{(t)},y)}\right) \]
3. With probability $\alpha(x^{(t)},y)$ accept the proposal, otherwise keep the old value, $x^{(t)}$:
   3.1 Draw $u \sim \mathcal{U}(0,1)$.
   3.2 Take
   \[ x^{(t+1)} = \begin{cases} y, & \text{if } u < \alpha(x^{(t)},y) \\ x^{(t)}, & \text{if } u \geq \alpha(x^{(t)},y) \end{cases} \]

Burn-in and mixing

Unsuitable proposal distributions may lead to bad mixing, which in turn causes problems for the MCMC algorithm, as a result the chain may:

- fail to converge altogether.
- take very long to converge.
- exhibit high autocorrelation, implying that many samples are needed to ensure good estimates.

Another problem is that the Markov chain usually takes a number of iterations to converge. This period is called the burn in of the chain and should be discarded.
Gibbs sampling & The slice sampler

**Gibbs sampling:**
1. Choose a starting value \( z^{(0)} \).
2. Repeat for \( i = 1, \ldots, N \):
   i.1 Draw \( z_1^{(i)} \) from \( f(z_1 | z_2^{(i-1)}, \ldots, z_m^{(i-1)}) \).
   i.2 Draw \( z_2^{(i)} \) from \( f(z_2 | z_1^{(i)}, z_3^{(i-1)}, \ldots, z_m^{(i-1)}) \).
   : 
   i.m Draw \( z_m^{(i)} \) from \( f(z_m | z_1^{(i)}, \ldots, z_{m-1}^{(i)}) \).
3. \( z^{(1)}, z^{(2)}, \ldots, z^{(N)} \), is now a sequence of dependent draws approximately from \( f \).

**Slice sampling:**
Start with a value \( x^{(0)} \), and for \( t = 1, \ldots, T \) do:
1. Sample \( y^{(t)} | x^{(t)} \in U (0, f(x^{(t)})) \).
2. Sample \( x^{(t+1)} | y^{(t)} \in U \{ x : f(x) \geq y^{(t)} \} \)
This results in samples \( x^{(t)} \) from a Markov chain with stationary distribution \( f(x) \).

Monte Carlo integration

- Realize that many quantities can be written as expectations.
  - Expectations: \( E(X) = \int x f(x) \, dx \)
  - Probabilities: \( P(X > 2) = E(I(x > 2)) \).
  - Standard integrals:
    \[
    \int_0^{2\pi} \sin(2\pi/(x - \pi)^2) \, dx = 2\pi E_{U(0,2\pi)} \left( \sin(2\pi/(x - \pi)^2) \right).
    \]

- Use the Law of Large Numbers to approximate the expectation with an average.

**Algorithm:**
1. Draw \( N \) values \( x_1, \ldots, x_N \) from \( f \).
2. Approximate \( \tau = E(\phi(X)) \) with
\[
\tau_N = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i).
\]

Importance sampling

- Hard to sample from \( f(x) \).
- If we want to calculate \( E(\phi(x)) \), where \( \phi(x) \) and \( f(x) \) are dissimilar.

The basis of importance sampling is to rewrite the integral

\[
E_\phi(\phi(x)) = \int \phi(x) f(x) \, dx = \int \frac{\phi(x) f(x)}{g(x)} \, g(x) \, dx = E_g \left( \frac{\phi(x) f(x)}{g(x)} \right) = E_g (\phi(x))
\]

**Algorithm:**
1. Draw \( N \) values \( x_1, \ldots, x_N \) from \( g \).
2. Approximate \( \tau = E_\phi(\phi(x)) \) with
\[
\tau_N = \frac{1}{N} \sum_{i=1}^{N} \frac{\phi(x_i) f(x_i)}{g(x_i)}.
\]

Stochastic modeling

- Observed data \( y \).
- The data is assumed to be a sample from a random variable \( Y \) with distribution a \( P_0 \).
- The statistical model is a set of probability distributions, \( P \) that is assumed to contain \( P_0 \).
- The largest possible model would be \( P = \{ \text{all possible distributions that could generate} \, y \} \).
- Commonly we restrict the set of distribution to a come from a parametric class. Ex:
  \[
P_\theta = \{ \text{all Normal distributions with} \, \mu = \theta \, \text{and} \, \sigma^2 = 1 \}.
\]
Bayesian modeling

- $\theta$ is seen as a random variable, and inference is based on the posterior distribution $p(\theta|y)$.
- Bayes’ formula is used to calculate the posterior:
  \[
p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int_{\theta'} p(y|\theta')p(\theta') d\theta'} \propto p(y|\theta)p(\theta)
  \]
- $f(y|\theta)$ is the likelihood that describes how the data $y$ behaves conditionally on the parameters $\theta$.
- $f(\theta)$ is called the prior distribution and summarizes our prior information about $\theta$ before observing $y$.
- Some options:
  - Conjugate priors.
  - Improper (flat) priors.
  - Jeffrey’s prior.

Bayesian computation

Given the posterior $p(\theta|y)$, we want to make statements about properties of the posterior:

- Marginal distribution
  \[
p(\theta_1|y) = \int \cdots \int p(\theta_1, \ldots, \theta_m|y) d\theta_2, \ldots, \theta_m.
  \]
- Expected parameter values $E(\theta|y)$.
- Predictive distribution, $p(z|y) = \int p(z|\theta)p(\theta|y) d\theta$.
- etc.

Most of these are expectations with respect to the posterior and often not solvable analytically. Thus we will need Monte Carlo integration.

- A 95% credible or posterior probability interval contains $\theta$ with a probability of 95%, given observations.
- Hypothesis tests are done by studying $P(H_0|\text{data})$.

Frequentist inference

- As a frequentist one assumes that there is a fixed true value $\theta_0$ for the parameters in the distribution $P_0$.
- Given the observations $y$, we want to make inference about some property, $\tau(P_0)$.
- We want to find a statistic $t(y)$ that is a good estimate, e.g. using Maximum likelihood or Least squares.
- To assess the uncertainty of this estimator we must then find the error distribution $F_\Delta$ of $\Delta(Y) = t(Y) - \tau$.
- In some cases we can derive this distribution analytically or find asymptotic estimates.
- Otherwise we can use Bootstrap to estimate it.

Bootstrap

**Principle:**

The population is to the sample as the sample is to the bootstrap samples.

$$t(Y) - \tau(P_0) = \mathcal{P} \approx t(y^*) - \tau(\mathcal{P})$$

**Algorithm:**

1. **Estimation:** Use data $y$ to construct an estimate $\hat{P}$ of $P_0$.
2. **Simulation:** Draw $B$ independent samples $y^{*b}$, $b = 1, \ldots, B$ from the distribution $\mathcal{P}$.
3. **Approximation:** Compute $t^{*b} = t(y^{*b})$, $b = 1, \ldots, B$, and use these values to approximate a plug-in estimate.
Hypotheses testing

The basis of a hypothesis test consists of

- A **null hypothesis** that we wish to test.
- A **test statistic** $t(y)$, i.e. a function of the data.
- A **rejection** or critical region $R$.

If the test statistic falls in the rejection region, then we reject the null hypothesis. We divide hypotheses into

**Simple** Specifies a single distribution for the data, e.g. $y \in N(\theta, \sigma^2)$ with $H_0: \theta = 0$ and $\sigma^2$ known.

**Composite** Specifies more than one distribution for the data, e.g. $y \in N(\theta, \sigma^2)$ with $H_0: \theta = 0$ and $\sigma^2$ unknown; or $y \in N(\theta, \sigma^2)$ with $H_0: \theta \leq 0$ and $\sigma^2$ known.

Computer Exercise 5

Let's look at an example with artificial data:

$$Y_1(t) = [1 \ 2 \ \cdots \ 20 \ 21 \ 50 + 5t],$$
$$Y_2(t) = [-t \ 23 \ 24 \ \cdots \ 41 \ 42 \ 60 + 6t].$$

with $t = 0, 1, \ldots, 99$. Note that $\bar{Y}_1(t) - \bar{Y}_2(t)$ is constant.

Computer Exercise 5

- First task was to illustrate that you can Bootstrap not only parameters and expectations.
- Second task was a comparison of three different test.
- The results are very different.
- This illustrates the fact that in hypothesis testing we have to consider:
  - What question are we asking?
  - What are the potential problems with the model?
  - What does the answer mean?
The EM-algorithm

Basic setup:
- We have observed some data $y$.
- Additional data $z$ is “missing”.
- The estimation problem would be “easy” if $z$ was known.

In principle we could write out the likelihood for the observed data as

$$L(\theta; y) = \int f(y, z|\theta) \, dz.$$ 

However the integral over the unknown data is often hard to compute. Instead we want to construct an approximation of the log-likelihood based on the observed data.

Algorithm:
Choose a starting value $\theta^{(0)}$.

E-step
Compute the expectation of the log-likelihood with respect to the unknown data, conditional on the parameter guess and known data.

$$Q(\theta, \theta^{(i-1)}) = E_z \left( \log L(\theta; y, z) | y, \theta^{(i-1)} \right).$$

M-step
Compute the maximum with respect to the unknown parameter

$$\theta^{(i)} = \arg \max_{\theta} Q(\theta, \theta^{(i-1)}).$$

Repeat until convergence.

INLA

- Lately alternatives to MCMC have been explored.
- A special case is an underlying Gaussian field with pointwise observations.
- In this case it turns out that clever programming and few hyper-parameters gives a situation that avoids MCMC.
- The result is the Integrated Nested Laplace approximation (INLA).
- An R-package exists.

- Given a model with an underlying Gaussian density $X \sim N(\mu(\theta), \Sigma(\theta))$ and pointwise observations $Y_i | X \sim g(\theta, X_i)$.
- The posterior of the parameters is

$$p(\theta | y) = \frac{p(x, y, \theta)}{p(x, y)} \propto \frac{p(y | x, \theta) p(x | \theta) p(\theta)}{p(x | \theta, y)}.$$ 

- Approximate $p(x | \theta, y)$ with a Gaussian, $p_G(x | \theta, y)$, centred at the mode $x^*(\theta)$.
- Use numeric optimisation and integration to find $p(\theta | y)$.
- Compute $p(x | y)$ as a weighted sum of Gaussians

$$p(x | y) = \int p(x | \theta, y) p(\theta | y) \, d\theta \approx \sum_{\theta} p_G(x | \theta, y) p(\theta | y).$$
Coal mine

\[ y_i|x_i \sim \text{Po}(\exp(x_i)), \]
\[ x \in \text{AR}(1). \]

Some Matlab tips

- When implementing programs in Matlab there are many things one can do to improve the computational efficiency.
- To measure how long it takes to run a program you can use `tic;` and `toc;`.
- To see which parts of your program that takes a long time to run you can use the profiler in Matlab:

  ```matlab
  %Turn on the profiler:
  profile on
  %Run your program
  answer = your_program(in_parameters);
  %Turn off the profiler and view the results:
  profile off;
  profile viewer;
  ``

Example: Challenger data

- For the Challenger data in the second home assignment, the log likelihood was

  \[ \log p(y|\theta) = \sum_{i=1}^{n} \log(p(y_i|\theta)) \]

  where \( p(y_i|\theta) \) is the \textit{Bin}(1,f(t)) density evaluated in \( y_i \).

- In matlab we can calculate this as

  ```matlab
  li = zeros(1,length(Y));
  for i = 1:length(Y)
    li(i) = log(binopdf(Y(i),1,1/(1+exp(-theta(1)-theta(2)*t(i)))));
  end
  l = sum(li);  
  ```

  10000 evaluations now only takes 4.4 seconds.

Example: Challenger data

- Instead calculate the logarithm directly and simplify as far as possible, also pre-allocate memory for the vector.

  ```matlab
  li = zeros(1,length(Y));
  for i = 1:length(Y)
    li(i) = -n*log(1+exp(-theta(1)-theta(2)*t(i))) + (n-Y(i))*(-theta(1)-theta(2)*t(i));
  end
  l = sum(li);
  ```

  10000 evaluations now only takes 4.4 seconds.
Example: Challenger data

- Looking at the profiler results again we see that most of the time still is spent in the for-loop.
- Matlab is often much more efficient when we vectorize the code:
  
  ```matlab
  l = -n* log(1+exp(-theta(1)-theta(2)*t)) + (n-Y).*(-theta(1)-theta(2)*X);
  l = sum(l);
  ```

- 10000 evaluations now only takes 2.8 seconds. With a few simple changes, our program is now about 25 times faster.

Oral exam

- Possible times to take the oral exam are:
  - Friday 11/3
  - Monday 14/3
  - Monday 21/3
  - Tuesday 22/3
  - Wednesday 23/3
  - You sign up for a time using Doodle, you will find the link on the homepage.
  - If you have not passed some computer lab, make sure to bring your code to the labs this week.
  - I will continue to have office hours at Mondays and Wednesdays until the last project is handed in.
  - If you cannot make it to the office hours, send me an email to make an appointment.

Further courses

- FMS155 Statistical Modelling of Extreme Values  
  - VT11 LP4
- FMS150 Statistical image analysis  
  - HT11 LP1
- FMSF05 Probability theory  
  - VT11, LP3
- FMSN10 Survival analysis  
  - HT11, LP2
- FMS170 Valuation of Derivative Assets  
  - HT11, LP1 & 2
- FMS072 Design of Experiments  
  - VT11, LP4
- FMS110 Non-linear Time Series Analysis  
  - HT11, LP1 & 2
- FMS161 Financial Statistics  
  - HT11, LP2

Master’s thesis

There are many different areas in statistics you can focus on in a Master’s thesis. Here are a few active research areas at the department:

- Nonparametric inference  
  - Dragi Anevski, Jimmy Olsson
- Mathematical Finance  
  - Erik Linström, Magnus Wiktorsson
- Spatio-temporal Stochastic Models  
  - Johan Lindström, David Bolin
- Statistical Signal Processing  
  - Andreas Jakobsson, Maria Sandsten
- Stochastics in Medicine  
  - Maria Sandsten, Anna Lindgren